



'Thrive in the Gift of Life'

Farington Moss St Paul's CE Primary School
Computing Scheme of Work
Computer Science

July 2024

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

The St Paul's scheme of work for Computer Science is a bespoke curriculum that uses resources from units in the Purple Mash scheme of work and the NCCE Teach Computing curriculum. Other physical resources e.g. Lego Spike, Kubo, Bee Bots and BBC Micro bit are also used to enhance the curriculum. The units from Purple Mash and the NCCE give teachers frameworks to plan lessons from and examples of activities and skills to teach. Once children have had opportunities to learn and tinker with new skills, they should create an end of unit outcome which demonstrates the skills in the unit of work.

For each Computer Science unit, pupils' learning and understanding must be supported by using different degrees of scaffolding, when teaching pupils to code. When planning each unit of work, teachers should provide children with many levels of scaffolding to clarify their thinking and practise their skills. The end of unit outcome and the project planning and evaluation can form part of the teacher assessment. It is important that children are taught and can use the topic vocabulary for each unit. Use retrieval activities to help consolidate children knowledge.

Scaffolding	Task Type	Examples of how to provide these opportunities
Most scaffolded 	Copying code	Give the children examples of code to copy.
	Targeted tasks	Read and understand code. Remix code to achieve a particular outcome. Debug code. Children use printed snippets of code to learn to read it. Unplugged activities to explain tasks.
	Shared coding	Share challenge activities as a class or group on the whiteboard. Complete a guided activity as a class. Share methods to create a class challenge. Use activities/examples from Purple Mash.
	Guided exploration	Explore a limited repertoire of commands. Remix code. Explore commands in free code before being taught what they do. Use questioning to support children's learning. Use PRIMM approach: Predict-Run-Investigate-Modify-Make
	Project design and code	Projects (imitate, innovate, invent, remix) Use example projects from the 2Code activities and the challenges at the end of each guided activity to help scaffold children's learning. Use free code to create a project that imitates a high-quality example.
Least scaffolded	Tinkering	Use free code to explore freely. Children complete an end of unit outcome.

(Adapted from Purple Mash and work by Jane Waite - Computing at Schools)

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

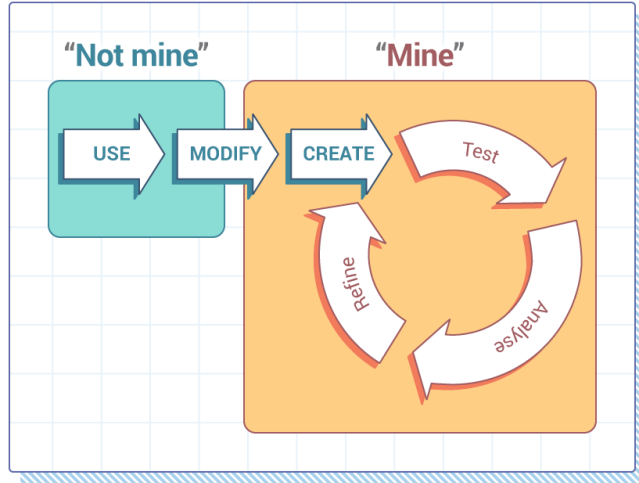
Programming Frameworks

When teaching a programming concept, there are two useful frameworks to use to ensure learners are not overwhelmed with the concepts being taught. These frameworks are suggested in the NCCE Teach Computing Curriculum. The first framework is called levels of abstraction. When programming, there are four levels that can help describe a project. Research suggests that this structure can support learners in understanding how to create a program and how it works:

- Task – what is needed
- Design – what it should do
- Code – how it is done
- Running the code – what it does

Spending time at the 'task' and 'design' levels before engaging in writing code aids learners in assessing the achievability of their programs and reduces the cognitive load for learners during programming.

The second framework is called Use-Modify-Create. The learner uses a program first before modifying it to create their own program. The diagram below explains this process.



(Taken from NCCE)

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Program Design

When a child is ready to design their own program, it is important for them to plan what the program is going to do. A blank design template is at the end of this document. It can be modified to suit the needs of the project being planned. These are some examples of design templates that could be used.

Example of a program design sheet

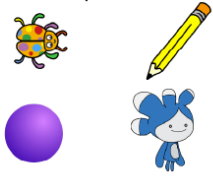
Task	Background	Sprite	Algorithm
Get a fish to swim from one side of the screen to the other and back again			1.  → 16 steps 2.  ← 16 steps 3. Stop

(Taken from NCCE Teaching Programming 5-11 year olds)

Design templates can be scaffolded for the children to complete. Different amounts of information can be added to the design template to support children according to their need. This type of scaffolding can be very useful for supporting children with SEND. This support will enable children to create an effective design programme within given parameters.

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

An example of a scaffolded design template.

Artwork	Algorithm
Choose a sprite: 	When the program starts Move the ball to the start position Repeatedly check: If the up arrow key is pressed Change the sprite y position by 10 If the down arrow key is pressed Change the sprite y position by -10 If the right arrow key is pressed Change the sprite x position by 10 If the left arrow key is pressed Change the sprite x position by -10
Backdrop: 	

(Taken from NCCE Teaching Programming 5-11 year olds)

Debugging

Teaching children to fix their errors with their coding is important.

Children should

1. Review the task. Look back to the design stage.
2. Test the program. Run the program to look for bugs in the code.
3. Identify the error. Look at each line of code.
4. Fix the bug.
5. Test the program again.

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Coding Skills

Year Group	Programming Skills Taught
EYFS	Introducing instructional language Children to be 'in role' giving instructions Experiment with playing with giving a robot a set of instructions
Year 1	Sequence Directional Language using robot step by step instructions then extend to creating a sequence of instructions. Create a simple programme. Add a background and an object(sprite). Make the sprite move.
Year 2	Sequence Recap creating a sequence of instructions using directional language. Begin to understand the term debugging and look for ways to improve algorithms. Programme a sequence of instructions into a robot. Create a programme with a background and multiple sprites moving.
Year 3	Repetition Recap directional language. Begin to understand the terms inputs and outputs. Creating a sequence of code. Use repetition.
Year 4	Selection Recap what inputs and outputs are. Create a sequence of code using repetition. Introduce selection - if/else statements.
Year 5	Function/Variables Create sequence of code Use selection - if/else statements. Introduce functions and variables. Introduce decomposition.
Year 6	Variables Recap functions, variables and selection. Create a sequence of code using selection (if/else statements) and variables to create a game.

EYFS

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

EYFS Computer Science Throughout the Year	
Early Learning Goals	Early Learning Goals: PD - Use a range of small tools PSED - Be confident to try new activities and show independence, resilience and perseverance in the face of challenge Development Matters: PD - Develop their small motor skills so that they can use a range of tools competently, safely and confidently. PSED - Show resilience and perseverance in the face of challenge.
End of Unit Outcomes	Look around the school and environment at technology with control switches, (e.g. photocopier, alarms, washing machines, television sets). Online simulation games - BBC website C Beebies. Talk about electronic equipment in real life situations, (e.g. traffic lights, scanners, microwaves, cash tills, etc.) and investigate how they work Encourage exploratory play with remote control toys, webcams, onscreen activities and simple control devices (e.g. microwave, kettle vacuum etc.)
Key Thread	Computer Science

Year 1

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 1 Computing Spring Term 1	
National Curriculum Objectives	- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions - Create and debug simple programs - Use logical reasoning to predict the behaviour of simple programs
End of Unit Outcomes	- Understand what algorithms are and develop strategies to help find bugs in them - Make a simple set of instructions to make a physical robot move using directional language
Key Thread	Computer Science/Computational Thinking
Prior Learning	EYFS
Intent	- Give commands and follow instructions - Plan, generate and follow a set of instructions - Identify errors in instructions
Sequence of Learning	- Practical teacher/children in role giving and following instructions (algorithms) e.g. building Duplo or Lego or another toy, paint by numbers (Purple Mash resources). - Children understand directional language - forwards, backwards, turn left, turn right - Experience of programming a set of instructions into Bee Bots/ Kubo around a mat. Initially step by step progressing, if possible, to multiple steps of instructions. - Children will create a programme and test it out on a robot. Debug any errors. - Children will plan a route and write a program for the route using directional language (end of unit outcome).
Vocabulary	Bee-Bot, Kubo, forwards, backwards, turn, clear, go, commands, instructions, directions, left, right, route, plan, algorithm, program, debug.
Resources	Purple Mash Unit 1.4 Examples of following instructions to build Logo Models, make a sandwich, paint by numbers Kubo lesson plans and Kubo sets Purple Mash 2Go - practise using directional language to navigate mazes NCCE Teach Computing Unit Year 1 Unit 3 Programming a Robot Bee Bot and mats
Retrieval/Assessment/ End of Unit Outcome	Use quizzes to check children's understanding of the topic language and screen shots or photographs of buttons on Bee Bots or tiles from Kubo. Assess children on planning and executing a route to program with a Bee Bot or Kubo.

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 1 Computing Summer Term 2	
National Curriculum Objectives	- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions - Create and debug simple programs - Use logical reasoning to predict the behaviour of simple programs
End of Unit Outcomes	- Understand what algorithms are and develop strategies to help find bugs in them. - Make very simple programs that follows an algorithm.
Key Thread	Computer Science/Computational Thinking
Prior Learning	Year 1 Spring 1
Intent	- Give commands and follow instructions - Plan, generate and follow a set of instructions - Identify errors in instructions - To understand the importance of precision in algorithms and programs - To use logical reasoning to predict what will happen - To plan and design a simple program using a background and a character(s) that move.
Sequence of Learning	Recap directional language used to program a robot. Children practise in role as programmer and robot. Understand the term algorithm Look at examples of code in projects. Learn how the projects and blocks of code are put together. Look at how to set up backgrounds and sprites(objects). Children to program an algorithm for a sprite to move using directional language. Children will design and write a programme with a background and a sprite. Children will programme the sprite to move.
Vocabulary	command, sprite (object), programming, area, block, joining, start, run, program, background, delete, reset, algorithm, predict, effect, change, value, instructions, design, event, debug.
Resources	Use ideas from Purple Mash Unit 1.7. Examples of coding projects Fun with Fish, Bubbles, Air Traffic Control. Use Free Code Chimp for the children's own project. Ref Purple Mash Unit 1.7 Learning Mat.
Retrieval/Assessment/ End of Unit Outcome	Use quizzes to check children's understanding of the topic vocabulary. Screen shot blocks of code for children to read/debug. Assess children's independent project.

Year 2

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 2 Computing Spring Term 1	
National Curriculum Objectives	- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions - Create and debug simple programs - Use logical reasoning to predict the behaviour of simple programs
End of Unit Outcomes	- Use algorithms and know that they can be implemented as programs on devices. - Know what debugging is and find errors in their programs. - Understand that programs execute by following a precise set of instructions. - Create simple programs and further develop their strategies and logical thinking to find bugs and predict outcomes in their algorithms and programs.
Key Thread	Computer Science/Computational Thinking
Prior Learning	Year 1 Spring Term 1 - understanding directional language and programming a robot. Creating a simple programme using directional language.
Intent	- Give commands and follow instructions - Plan, generate and follow a set of instructions - Begin to Identify errors in instructions (debugging) - To understand the importance of precision in algorithms and programs - To use logical reasoning to predict what will happen - To understand prediction, trial and error are important considerations when creating programs
Sequence of Learning	Children to recap in role their understanding of directional language - forwards, backwards, left and right. Children to recap the importance of giving and receiving a precise set of instructions (step by step and as a sequence). Introduce the term algorithm. Children to practise these skills initially with an unplugged activity. Give children opportunities to practise debugging skills. Children to look at algorithms that need improving. Children to create a sequence of instructions for a programmable robot to move around a map. Use programmable toys (Kubo) to look at debugging instructions (ref Kubo lesson plans for more information).
Vocabulary	instruction, sequence, clear, unambiguous, algorithm, program, order, prediction, design, route, map, debugging, decomposition
Resources	Kubo Kubo lesson plans Barefoot Computing (ideas for unplugged activities) https://www.barefootcomputing.org/primary-computing-resources NCCE Teach Computing Unit Year 2 Unit 3 Programming a Robot
Retrieval/Assessment/End of Unit Outcome	Use quizzes to check children's understanding of the topic language and screen shots or photographs of tiles from Kubo. Assess children planning and executing a route on Kubo.

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 2 Computing Summer Term 2	
National Curriculum Objectives	- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions - Create and debug simple programs - Use logical reasoning to predict the behaviour of simple programs
End of Unit Outcomes	- Use algorithms and know that they can be implemented as programs on devices. - Know what debugging is and find errors in their programs. - Understand that programs execute by following a precise set of instructions. Create simple programs and further develop their strategies and logical thinking to find bugs and predict outcomes in their algorithms and programs.
Key Thread	Computer Science/Computational Thinking
Prior Learning	Year 1 Summer Term 2
Intent	- Give commands and follow instructions - Plan, generate and follow a set of instructions - Identify errors in instructions(debugging) - To understand the importance of precision in algorithms and programs - To use logical reasoning to predict what will happen - To understand prediction, trial and error are important considerations when creating programs
Sequence of Learning	Recap directional language skills taught from the previous unit and Year 1. Recap what an algorithm is and the importance of precise instructions. Use unplugged activities, if needed. Look at examples of code in projects and learn how they are put together. Use a timer and learn how to move characters in different ways. Make multiple characters move in directions and collision detection. Use a timer to make characters hide/appear. Provide opportunities to debug algorithms. Create and plan a simple program using multiple characters, collision detection, timer, hide/appear.
Vocabulary	sequence, command, program, run, start, outcome, predict, blocks, design, actions, sprite, project, modify, change, algorithm, build, match, compare, debug, features, evaluate, decomposition, code.
Resources	Purple Mash Unit 2.1 Coding - Examples of Code Air Traffic control, Princess and the Frog, Magician. Use Free Code Chimp for the children's own project. Ref Purple Mash Unit 2.1 Learning Mat.

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Retrieval/Assessment/ End of Unit Outcome	Use quizzes to check children's understanding of the topic vocabulary. Screen shot blocks of code for children to read/debug. Assess children's independent project.
--	--

Year 3

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 3 Computing Autumn Term 2	
National Curriculum Objectives	- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts - Use sequence, selection, and repetition in programs; work with variables and various forms of input and output - Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
End of Unit Outcomes	- Plan and write algorithms and programs using sequence and repetition and further develop their computational thinking strategies to solve problems and errors in their algorithms and programs. - Have knowledge and experience of using a range of different input and outputs.
Key Thread	Computing Science/Computation Thinking
Prior Learning	Year 1 Spring 1, Year 2 Spring 2
Intent	To plan and write programs that accomplish specific goals To read what a sequence in a program does To create and debug programs that implement algorithms to achieve specific goals To use sequence and repetition To understand what a variable is in programming. To use logical reasoning to detect and correct errors in programs. To work with various forms of inputs and outputs To use sequences of commands to control physical devices using outputs To use and debug programs that control physical devices (outputs)
Sequence of Learning	Children need to understand what input and output devices are - (input) keyboards, mice, touch screens, microphones, cameras, (output) screens, printers, speakers. Check BBC Bitesize for further information. Look at different input and output devices Children plan and write programs with different outputs - motors, sensors, light blocks. Follow the Lego Spike lesson plans for the Great Adventures unit. Children use the last lesson The Great Desert Adventure as the end of unit outcome.
Vocabulary	algorithm, command, input, output, program, sequence, repetition, blocks, commands, code, run the code, bug, debug
Resources	Lego Spike Essential Sets, iPads, Lego Spike app - Great Adventures Unit, Lego Education website https://education.lego.com/en-gb/lessons/
Retrieval/Assessment/End of Unit Outcome	Use quizzes to check children's understanding of the topic vocabulary. Screen shot blocks of code for children to read/debug. Assess children's Great Desert Adventure project.

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 3 Computing Spring Term 2	
National Curriculum Objectives	- design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts - use sequence, selection, and repetition in programs; work with variables and various forms of input and output - use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
End of Unit Outcomes	- Plan and write algorithms and programs using sequence and repetition and further develop their computational thinking strategies to solve problems and errors in their algorithms and programs. - Have knowledge and experience of using a range of different input and outputs.
Key Thread	Computing Science/Computation Thinking
Prior Learning	Year 1 Summer Term 2, Year 2 Summer Term 2.
Intent	To plan and write programs that accomplish specific goals To use sequence in programs To read what a sequence in a program does To create and debug programs that implement algorithms to achieve specific goals To use sequence and repetition To understand what a variable is in programming. To use logical reasoning to detect and correct errors in programs.
Sequence of Learning	Recap directional language skills taught from the previous years. Recap what an algorithm is and the importance of precise instructions. Use unplugged activities, if needed. Look at examples of code in projects and learn how they are put together. Use a timer with an 'every x seconds' command and an 'after x seconds' command. Introduce the concept of repetition. Explore examples of repetition code. Provide opportunities to debug algorithms. Create and plan a simple program using multiple characters, collision detection, timer, hide/appear. Practise repetition by drawing shapes on. Learn how to use nesting (when a command is written within another command). Create a project that uses the skills of a timer, sequence of commands, repetition and nesting. Use Free Code Chimp. Children could create a project similar to the jumping monkey.
Vocabulary	action, algorithm, bug, command, debug, if, input, output, selection, variable, timer, sequence, nesting
Resources	Purple Mash Unit 3.1 lessons 1-4 Ref Purple Mash Unit 3.1 Learning Mat.
Retrieval/Assessment/ End of Unit Outcome	Use quizzes to check children's understanding of the topic vocabulary. Screen shot blocks of code for children to read/debug. Assess children's independent project.

Year 4

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 4 Computing Autumn Term 1	
National Curriculum Objectives	- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts - Use sequence, selection, and repetition in programs; work with variables and various forms of input and output - Use logical reasoning to explain how some simple algorithms work and to detect and correct
End of Unit Outcomes	- Design and write more complex algorithms and programs using sequence, repetition and selection. - Further develop their computational thinking to help debug their programs and design and solve problem tasks. - Develop their understanding of inputs and outputs further, demonstrating how they can use programs to control external devices such as sensors, motors and robots.
Key Thread	Computing Science/Computation Thinking
Prior Learning	Year 1 Spring 1, Year 2 Spring 2, Year 3 Autumn 2
Intent	To plan and write programs that accomplish specific goals To read what a sequence in a program does To create and debug programs that implement algorithms to achieve specific goals To use sequence and repetition To use logical reasoning to predict outcomes To work with various forms of inputs and outputs To use sequences of commands to control physical devices using outputs
Sequence of Learning	Recap what an input and output device is from Year 3. Recap children's understanding of the term algorithms. Recap using precise instructions. Use unplugged activities if necessary. Discuss what a sensor is and how programs can control devices such as sensors and robots. Develop debugging skills. Plan, create a program and evaluate it to control an external device such as a robot using motors, sensors and light blocks. Use Lego Spike essential lesson plans. Complete the lesson plans for Quirky Creations. Children use the last lesson Your School Creation as an end of unit outcome. Cross curricular link to DT create a lampshade project. Children create and use algorithms in Crumble to light up a lampshade.
Vocabulary	algorithm, command, input, output, program, sequence, repetition, debug
Resources	Lego Spike Essential kits, iPads, Lego Spike app, Crumble kits, Lego Education website https://education.lego.com/en-gb/lessons/
Retrieval/Assessment/ End of Unit Outcome	Use quizzes to check children's understanding of the topic vocabulary. Screen shot blocks of code for children to read/debug. Assess children's Your school creation project.

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 4 Computing Summer Term 2	
National Curriculum Objectives	- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts - Use sequence, selection, and repetition in programs; work with variables and various forms of input and output - Use logical reasoning to explain how some simple algorithms work and to detect and correct
End of Unit Outcomes	- Design and write more complex algorithms and programs using sequence, repetition and selection. - Further develop their computational thinking to help debug their programs and design and solve problem tasks. - Develop their understanding of inputs and outputs further, demonstrating how they can use programs to control external devices such as sensors, motors and robots.
Key Thread	Computing Science/Computation Thinking
Prior Learning	Year 1 Summer Term 2, Year 2 Summer Term 2, Year 3 Spring 2
Intent	To read what a sequence in a program does To create and debug programs that implement algorithms to achieve specific goals To use sequence, selection and repetition To use logical reasoning to predict outcomes To work with various forms of inputs and outputs To use sequences of commands to control physical devices using outputs To use a sequence of commands to control physical devices using outputs.
Sequence of Learning	Recap what an algorithm is and the importance of precise instructions. Use unplugged activities, if needed. Look at examples of code in projects and learn how they are put together. Recap concept of repetition. Explore examples of repetition code. Introduce the concept of selection use an if statement and an if/else statement. Provide opportunities to debug algorithms. Create a simple program using an if statement to change the background. Create a project that uses the skills of a sequence of commands, repetition and an if statement. Use Free Code Chimp. Children could create an animal themed project (Science) using similar code to the It is raining, or Guard the Castle (Gibbon).
Vocabulary	algorithm, program, code, command, input, output, sequence, repeat, repetition, debug, selection, if statement
Resources	Purple Mash Unit 4.1 Lessons 1- 4. Ref Purple Mash Unit 4.1 Learning Mat.
Retrieval/Assessment/ End of Unit Outcome	Use quizzes to check children's understanding of the topic vocabulary. Screen shot blocks of code for children to read/debug. Assess children's independent project.

Year 5

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 5 Computing Spring Term 2	
National Curriculum Objectives	- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts - Use sequence, selection, and repetition in programs; work with variables and various forms of input and output - Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
End of Unit Outcomes	- Design and write programs using sequence, repetition, selection and variables. - Develop greater understanding of how to use selection and repetition in more complex programs. - Further develop their computational thinking showing they can plan and decompose tasks; explain how the algorithms they write work and correct errors in their programs. - Plan and write programs to control external devices such as sensors, motors and robots and explain about the inputs and outputs used.
Key Thread	Computing Science/Computation Thinking
Prior Learning	Year 1 Spring 1, Year 2 Spring 2, Year 3 Autumn 2, Year 4 Autumn 1
Intent	To use repetition, selection in programs To use variables in programs To design and create programs using decomposition To design programs to achieve specific goals To use logical reasoning to develop systematic strategies to debug algorithms
Sequence of Learning	Recap children's understanding of the term algorithms and the importance of precise instructions. Use unplugged activities if necessary. Plan, create a program and evaluate it to control an external device such as a robot using motors, sensors and light blocks. Use Lego Spike Essential lesson plans. Complete the lesson plans for Happy Traveller unit. Understand the term decomposition and apply the skills to the code used. Children to use the last lesson Get Around Town as the end of unit outcome.
Vocabulary	algorithm, decomposition, variable, abstraction, command, program, sequence, repetition, debug
Resources	Lego Spike Essential kits. iPads, Lego Spike app, Lego Education website https://education.lego.com/en-gb/lessons/
Retrieval/Assessment/End of Unit Outcome	Use quizzes to check children's understanding of the topic vocabulary. Screen shot blocks of code for children to read/debug. Assess children's Get Around Town creation project.

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 5 Computing Summer Term 2	
National Curriculum Objectives	- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts - Use sequence, selection, and repetition in programs; work with variables and various forms of input and output - Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
End of Unit Outcomes	- Design and write programs using sequence, repetition, selection and variables. - Develop greater understanding of how to use selection and repetition in more complex programs. - Further develop their computational thinking showing they can plan and decompose tasks; explain how the algorithms they write work and correct errors in their programs. - Plan and write programs to control external devices such as sensors, motors and robots and explain about the inputs and outputs used.
Key Thread	Computing Science/Computation Thinking
Prior Learning	Year 1 Summer Term 2, Year 2 Summer Term 2, Year 3 Spring 2, Year 4 Summer 2
Intent	To use repetition and selection in programs To use variables and functions in programs To design and create programs using decomposition To design programs to achieve specific goals To use logical reasoning to develop systematic strategies to debug algorithms
Sequence of Learning	Introduce the term variable. Look at the Day and Night Gibbon task on Purple Mash (Unit 4.1 Lesson 5). Look at examples of code with multiple variables. Learn how to simplify the code (animal race unit 5.1). Learn how to add speed to objects (catching game). Look at an example of a simulation (traffic lights). Understand the term decomposition. To understand the term function. Look how functions are used in the Football game. Children to create a game similar to the football game using variables and functions.
Vocabulary	algorithm, design, run, code, blocks, selection, variable, object, background, event, coordinates
Resources	Purple Mash Unit 4.1 Lesson 5, Purple Mash Unit 5.1 Lessons 1 - 4. Ref Purple Mash Unit 5.1 Learning Mat.
Retrieval/Assessment/ End of Unit Outcome	Use quizzes to check children's understanding of the topic vocabulary. Screen shot blocks of code for children to read/debug. Assess children's independent project.

Year 6

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 6 Computing Autumn Term 1	
National Curriculum Objectives	- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts - Use sequence, selection, and repetition in programs; work with variables and various forms of input and output - Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
End of Unit Outcomes	- Design and create more complex programs using sequence, repetition, selection and variables appropriately. - Develop their computational thinking can demonstrate that they can decompose and evaluate their tasks and correct errors in their algorithms and programs. - Be confident in their knowledge of inputs and outputs and plan and write programs to solve tasks to control external devices such as sensors and motors.
Key Thread	Computing Science/Computation Thinking
Prior Learning	Year 1 Summer Term 2, Year 2 Summer Term 2, Year 3 Spring 2, Year 4 Summer 2, Year 5 Summer 2, Year 6 Autumn 1
Intent	To introduce children to the Micro Bit To learn what a variable is To learn about inputs and outputs on a Micro Bit To plan, program and debug code using the Micro Bit
Sequence of Learning	Children will explore a BBC Micro Bit Children will learn what a variable is. Children will need to create a variable Children will explore the inputs and outputs on a Micro Bit Children will learn to use the input blocks of code Children will plan code to make a sports timer Children will debug code
Vocabulary	algorithm, action, command, debug, decomposition, event, input, object, output
Resources	NCCE Teach Computing Unit 6.7 Micro Bits Lesson 1 and 2 Additional Microbit Information on Purple Mash Unit 4.11
Retrieval/Assessment/ End of Unit Outcome	Use quizzes to check children's understanding of the topic vocabulary. Screen shot of Micro bits and blocks of code for children to read/debug/annotate. Assess children's independent project.

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 6 Computing Autumn Term 2	
National Curriculum Objectives	- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts - Use sequence, selection, and repetition in programs; work with variables and various forms of input and output - Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
End of Unit Outcomes	- Design and create more complex programs using sequence, repetition, selection and variables appropriately. - Develop their computational thinking can demonstrate that they can decompose and evaluate their tasks and correct errors in their algorithms and programs. - Be confident in their knowledge of inputs and outputs and plan and write programs to solve tasks to control external devices such as sensors and motors.
Key Thread	Computing Science/Computation Thinking
Prior Learning	Year 1 Summer Term 2, Year 2 Summer Term 2, Year 3 Spring 2, Year 4 Summer 2, Year 5 Summer 2
Intent	To use repetition and selection in programs To use variables in programs To design and create programs using decomposition To design programs to achieve specific goals To use logical reasoning to develop systematic strategies to debug algorithms
Sequence of Learning	Look at a game that uses a timer and if/else statements. Talk about how to simplify the code (Splatty bug) Recap using Functions to simplify the code so it is not repeated. Look at a simulation (Billy's Bedroom) Create a game in the style of Splatty bug, Reginald Rocket or a Text Adventure (Lesson 6) depending on the skill level of the children.
Vocabulary	algorithm, action, command, debug, decomposition, event, input, object, output
Resources	Purple Mash Unit 6.1 Lessons 1-4
Retrieval/Assessment/ End of Unit Outcome	Use quizzes to check children's understanding of the topic vocabulary. Screen shot blocks of code for children to read/debug. Assess children's independent project.

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

Year 6 Computing Summer Term	
National Curriculum Objectives	- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts - Use sequence, selection, and repetition in programs; work with variables and various forms of input and output - Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
End of Unit Outcomes	- Design and create more complex programs using sequence, repetition, selection and variables appropriately. - Develop their computational thinking can demonstrate that they can decompose and evaluate their tasks and correct errors in their algorithms and programs. - Be confident in their knowledge of inputs and outputs and plan and write programs to solve tasks to control external devices such as sensors and motors.
Key Thread	Computing Science/Computation Thinking
Prior Learning	Year 1 Spring 1, Year 2 Spring 2, Year 3 Autumn 2, Year 4 Autumn 1, Year 5 Spring 2
Intent	To use repetition and selection in programs To use variables in programs To design and create programs using decomposition To design programs to achieve specific goals To use logical reasoning to develop systematic strategies to debug algorithms To design, test and refine programs to control robots
Sequence of Learning	Recap children's understanding of the term algorithms and the importance of precise instructions. Use unplugged activities if necessary. Plan, create a program and evaluate it to control an external device such as a robot using motors, sensors and light blocks. Use Lego Spike Essential lesson plans. Complete the lesson plans for Amazing Amusement Park. Understand the term decomposition and apply the skills to the code used. Children to use the last lesson The Most Amazing Amusement Park as the end of unit outcome.
Vocabulary	algorithm, decomposition, variable, abstraction, command, program, sequence, repetition, debug
Resources	Lego Spike Essential kits. iPads, Lego Spike app, Lego Education website https://education.lego.com/en-gb/lessons/
Retrieval/Assessment /	Use quizzes to check children's understanding of the topic vocabulary. Screen shot blocks of code for children to read/debug. Assess children's independent project.

Farington Moss St Paul's CE Primary School Computer Science Scheme of Work

End of Unit Outcome	
---------------------	--

Computing Program Design Sheet

Task	Background	Sprite(s)	Algorithms

Example of a blank planning template